Singleton Pattern

2019.02.26.

Concept

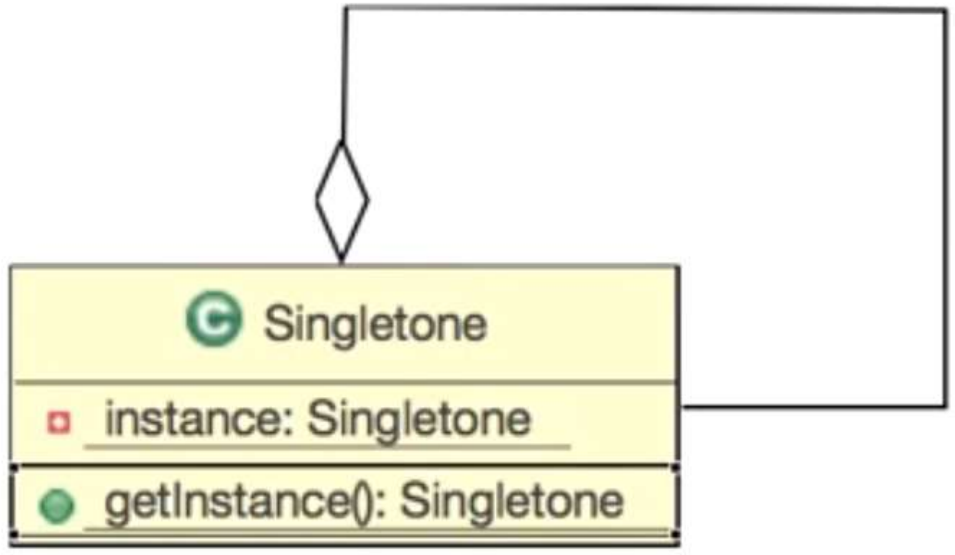
Object: With attributes and functions.

Class: Was defined attributes and functions

Instance: Having properties and functions that are real.

Goal of Lecture

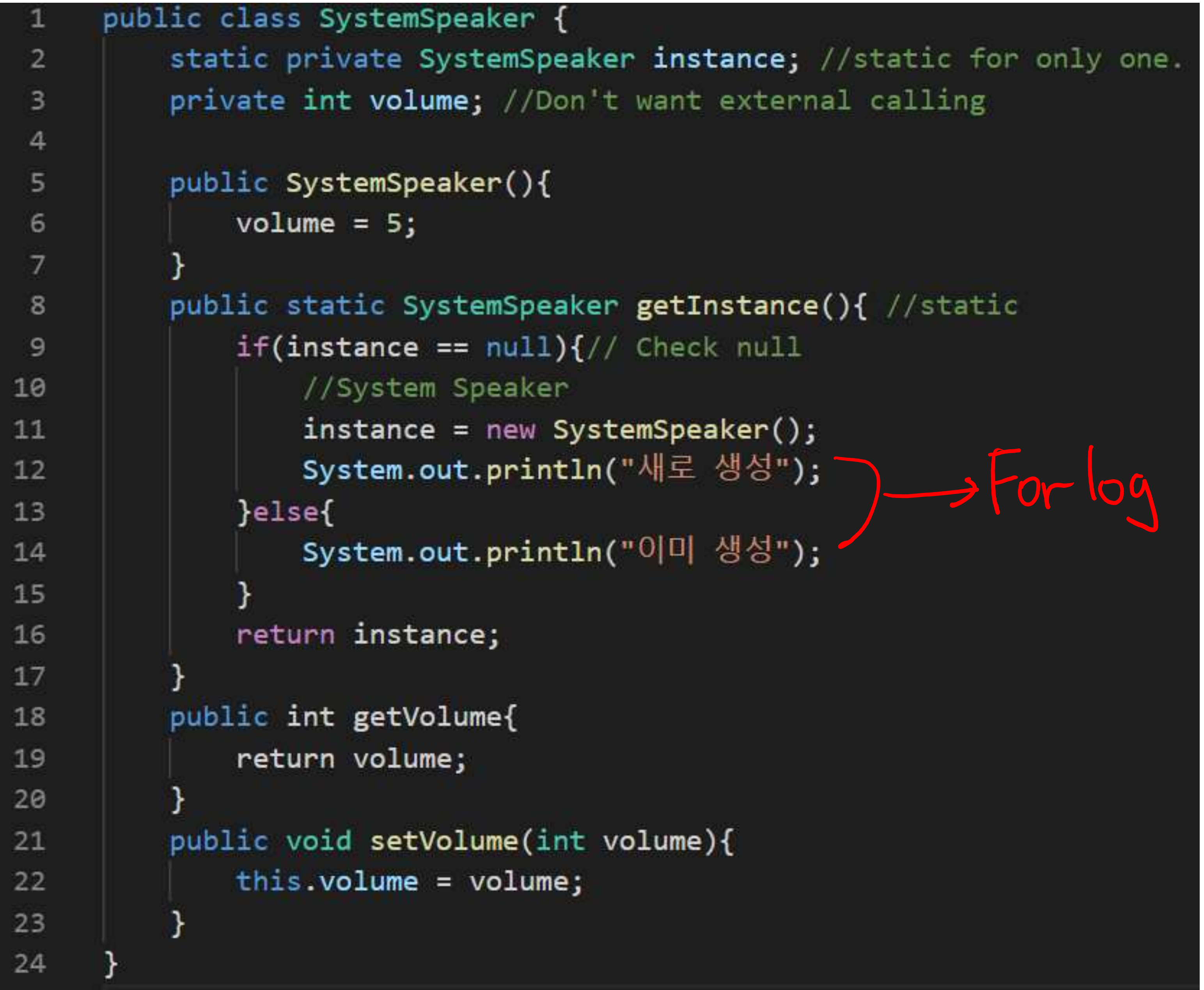
Implement to create only one instance through Singleton Pattern



Requirement

Make a class that can access the speaker in developing system.

SystemSpeaker.java



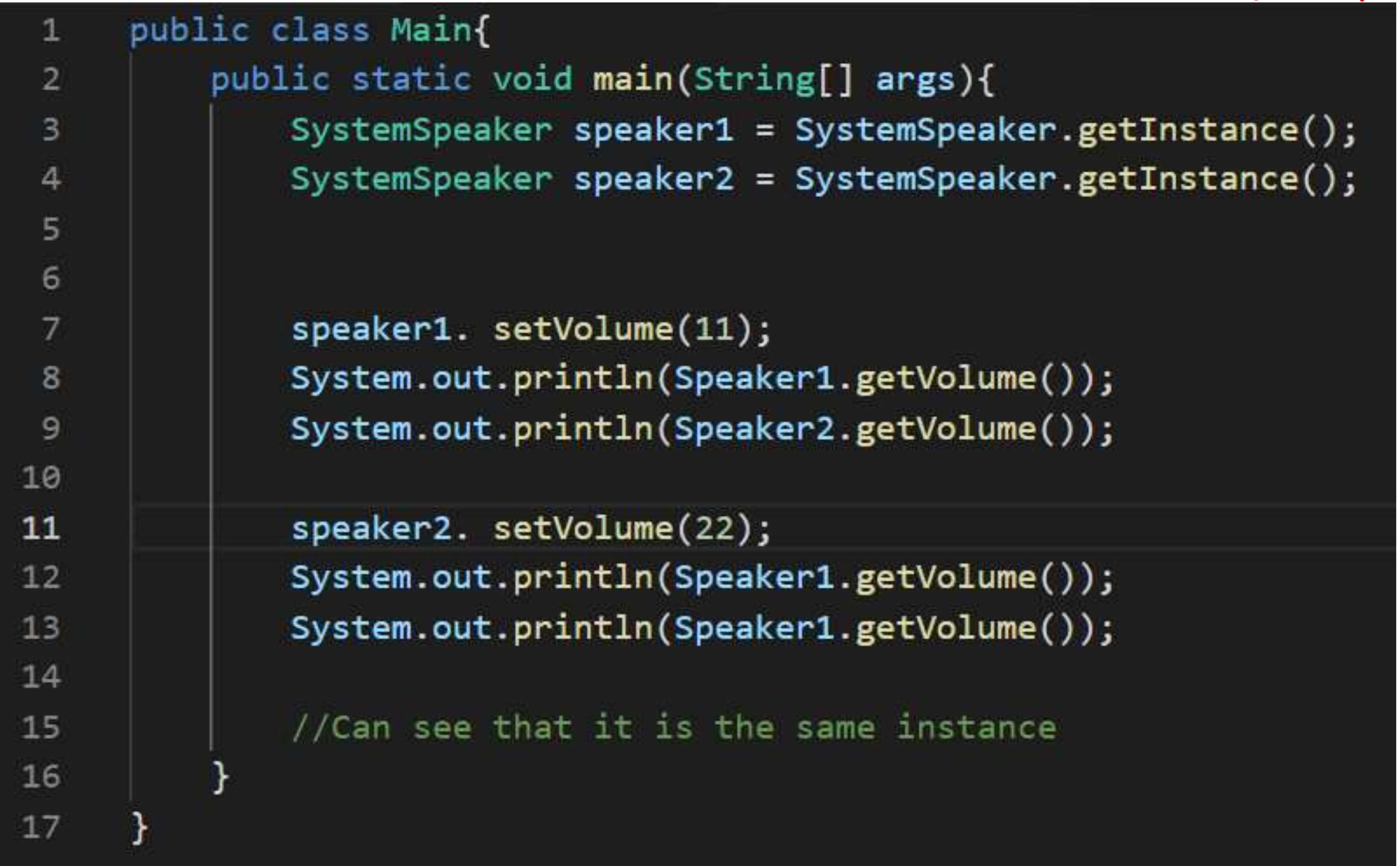
```java
public class SystemSpeaker {
    static private SystemSpeaker instance; //static for only one.
    private int volume; //Don't want external calling

    public SystemSpeaker(){
        volume = 5;
    }
    public static SystemSpeaker getInstance(){ //static
        if(instance == null){// Check null
            //System Speaker
            instance = new SystemSpeaker();
            System.out.println("새로 생성");
        }else{
            System.out.println("이미 생성");
        }
        return instance;
    }
    public int getVolume{
        return volume;
    }
    public void setVolume(int volume){
        this.volume = volume;
    }
}
```

For log

Main.java

Remember features of Singleton



```java
public class Main{
    public static void main(String[] args){
        SystemSpeaker speaker1 = SystemSpeaker.getInstance();
        SystemSpeaker speaker2 = SystemSpeaker.getInstance();


        speaker1. setVolume(11);
        System.out.println(Speaker1.getVolume());
        System.out.println(Speaker2.getVolume());

        speaker2. setVolume(22);
        System.out.println(Speaker1.getVolume());
        System.out.println(Speaker1.getVolume());

        //Can see that it is the same instance
    }
}
```